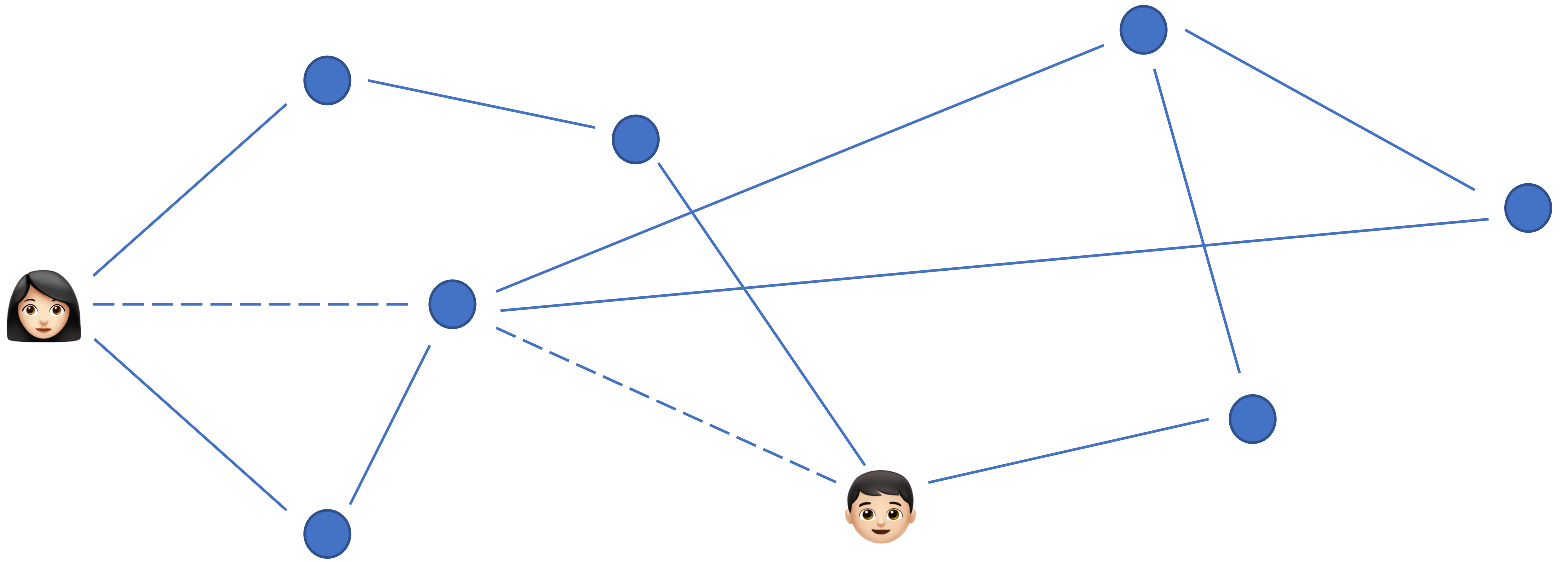
Wiser: Increasing Throughput in Payment Channel Networks with Transaction Aggregation

Samarth Tiwari[1], Michelle Yeo[2], Zeta Avarikioti[3], Iosif Salem[4], Krzysztof Pietrzak[2], Stefan Schmid[4]

[1]Centrum Wiskunde & Informatica Amsterdam, Netherlands
[2]Institute of Science and Technology Austria, Austria
[3]TU Wien, Austria
[4]TU Berlin, Germany

Background

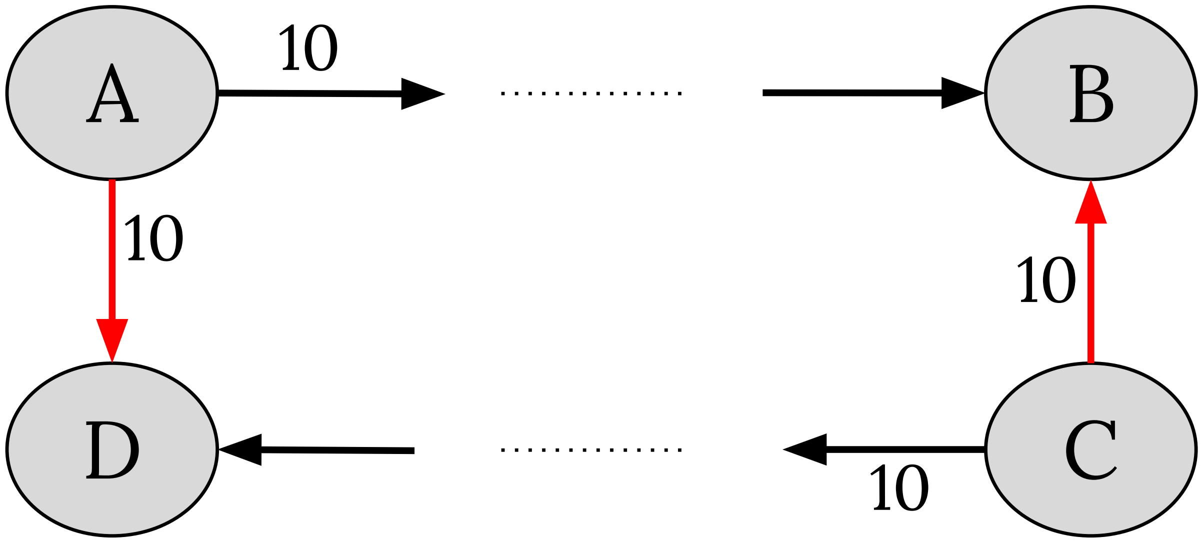
- PCNs: layer 2 solutions to improve scalability of blockchains
- Intermediary nodes on a payment path charge a routing fee



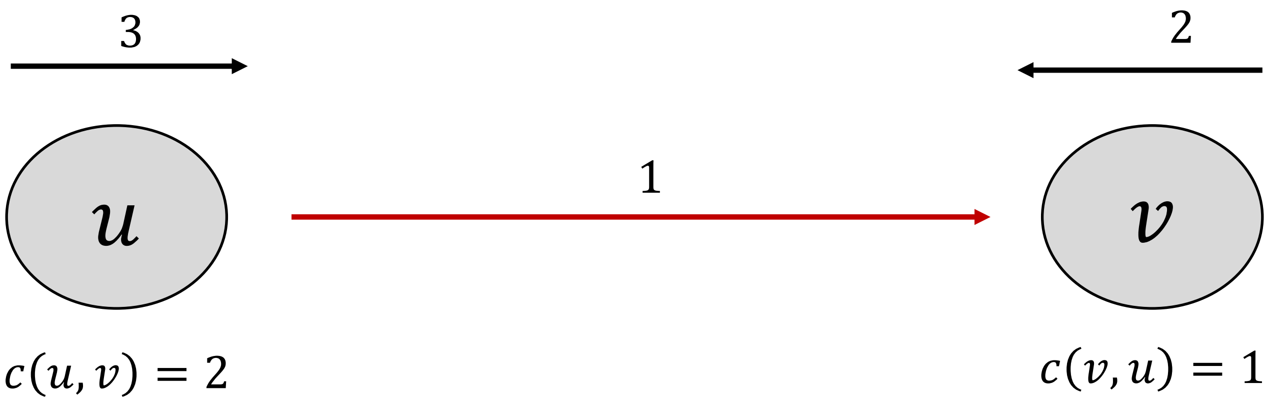
Transaction aggregation in PCNs

- Finding a set of channels which execute as many transactions as possible
  - Take into account both input transactions as well as the topology of the PCN
- Added benefit to users compared to sequential/individual execution

Motivating example 1



Motivating example 2



Our contribution

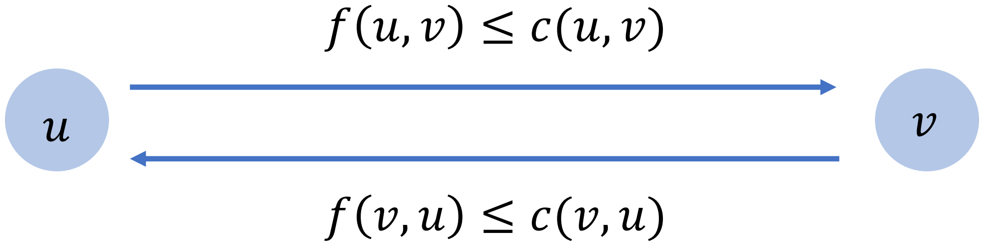
Wiser is the first solution that performs transaction aggregation in PCNs that satisfies the following properties:

1. Computational feasibility
2. Balance security
3. Optimality
4. Cost efficiency
5. Privacy

Computational problem definition

$G = (V, E), |V| = n, |E| = m$

- Flow vector f $= \big(f(e)\big)_{e \in E}, 0 \leq f(e) \leq c(e)\ \forall e \in E$



Computational problem definition

$G = (V, E), |V| = n, |E| = m$

- Flow vector f $= \left(f(e)\right)_{e \in E}, 0 \leq f(e) \leq c(e)\ \forall e \in E$
- List of transactions $T = \{t_1, t_2, \dots, t_k\}$

$$t_i = [0, \dots, w_{sender}, \dots, -w_{receiver}, \dots, 0]$$

- Demand vector $d = \sum_{t \in T} t$

Computational problem definition

$G = (V, E), |V| = n, |E| = m$

- Flow vector f $= \left(f(e)\right)_{e \in E}, 0 \leq f(e) \leq c(e)\ \forall e \in E$
- List of transactions $T = \{t_1, t_2, \dots, t_k\}$

$$t_i = [0, \dots, w_{sender}, \dots, -w_{receiver}, \dots, 0]$$

- Demand vector $d = \sum_{t \in T} t$
- A flow f routes d if $\forall\ v$,

$$\sum_{(v,u) \in E} f(v,u) - \sum_{(u,v) \in E} f(u,v) = d(v)$$

Computational problem definition

$G = (V, E), |V| = n, |E| = m$

- Flow vector f $= \left(f(e)\right)_{e \in E}, 0 \leq f(e) \leq c(e)\ \forall e \in E$
- List of transactions $T = \{t_1, t_2, \dots, t_k\}$

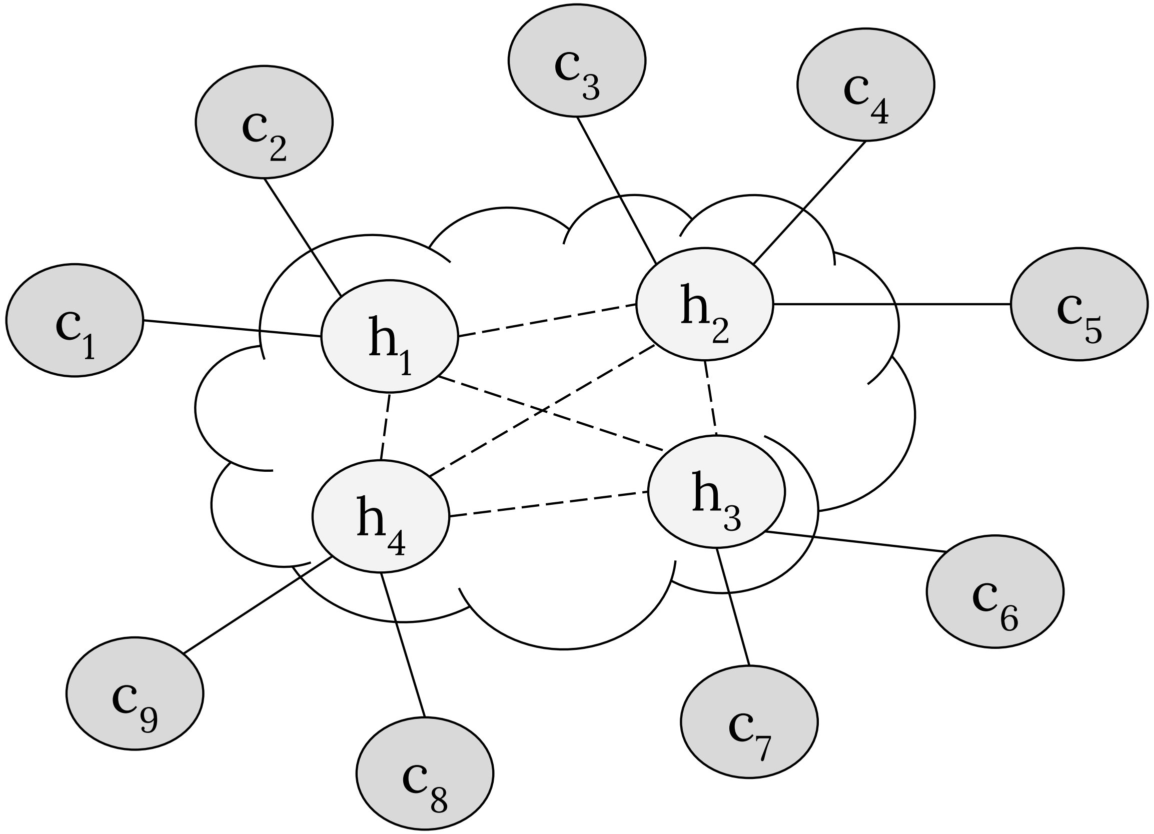
$$t_i = [0, \dots, w_{sender}, \dots, -w_{receiver}, \dots, 0]$$

- Demand vector $d = \sum_{t \in T} t$
- A flow f routes d if $\forall\ v,$

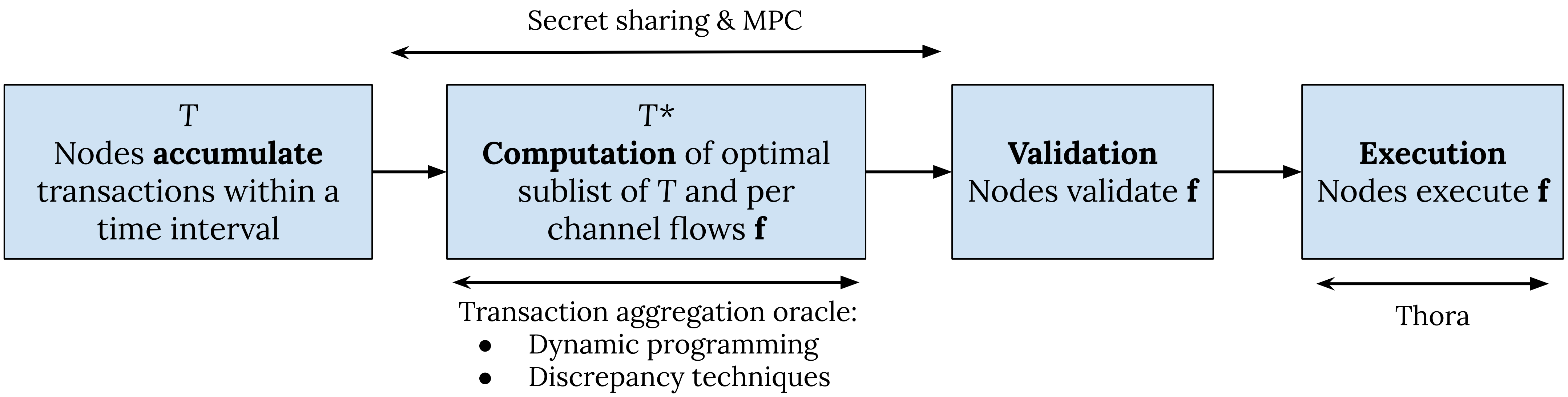
$$\sum_{(v,u) \in E} f(v,u) - \sum_{(u,v) \in E} f(u,v) = d(v)$$

Goal: find feasible subset $T' \subset T$ such that $\sum_{t_i \in T'} |t_i|$ is maximised

PCN model



Wiser protocol implementation



Secret sharing & MPC

T
Nodes **accumulate** transactions within a time interval

*T**
Computation of optimal sublist of *T* and per channel flows **f**

Validation
Nodes validate **f**

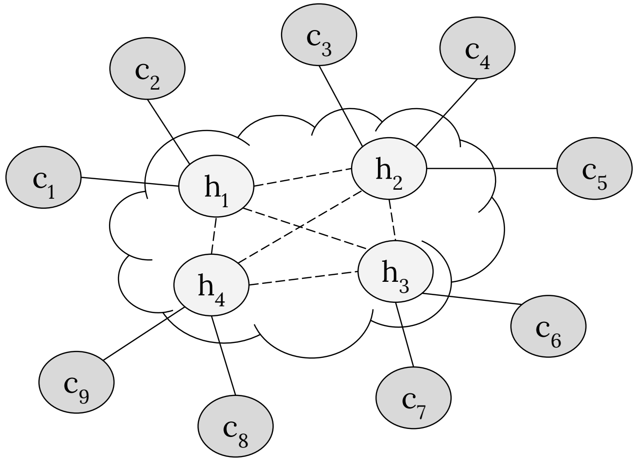
Execution
Nodes execute **f**

Transaction aggregation oracle:

- Dynamic programming
- Discrepancy techniques

Thora

Flow computation phase

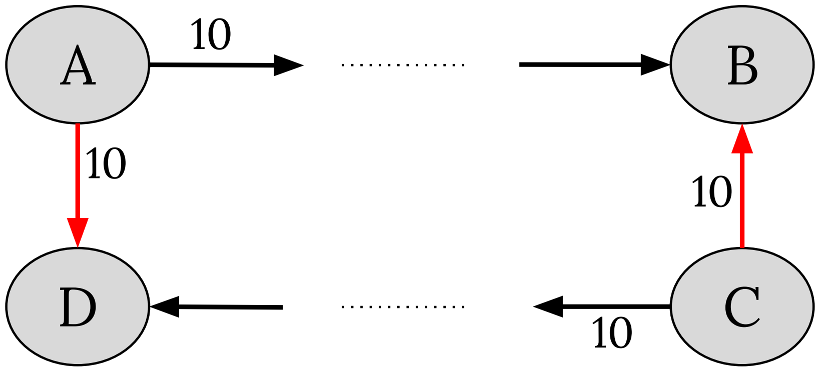


- MPC delegates sampled randomly from hubs
    - Prevents Sybil attacks
    - Sufficient computational and financial resources
- Secret sharing of transactions and channel balances to delegates
- Oracle to solve the computational problem
    - MPC so efficiency is important
    - Convert problem to integer program and use result by Eisenbrand and Weismantel[1] to solve in time linear in number of transactions and exponential in number of hubs.

[1] Eisenbrand and Weismantel, https://arxiv.org/abs/1707.00481

Atomic flow execution

- Typical HTLC based solutions require connectivity and locks payments for time linear in length of the path
- Flow output might involve disconnected components of network
- Thora[1]: multi-channel atomic updates in constant time



[1] Aumayr, Abbaszadeh, Maffei, https://eprint.iacr.org/2022/317

Analysis of Wiser

1. Computational feasibility
2. Balance security
3. Optimality
4. Cost efficiency
5. Privacy

Theorem: The transaction aggregation problem can be solved in time $O\left(k(h\Delta)^{h^2}\right)$

Analysis of Wiser

1. Computational feasibility
2. Balance security
3. Optimality
4. Cost efficiency
5. Privacy

Follows from atomic updates of Thora

Analysis of Wiser

1. Computational feasibility
2. Balance security
3. Optimality
4. Cost efficiency
5. Privacy

Follows from correctness of optimization solver

Analysis of Wiser

1. Computational feasibility
2. Balance security
3. Optimality
4. Cost efficiency
5. Privacy

Follows from the fact that fee function satisfies triangle inequality

Analysis of Wiser

1. Computational feasibility
2. Balance security
3. Optimality
4. Cost efficiency
5. Privacy

Follows from security guarantees of the MPC protocol

Conclusion

- First solution that performs transaction aggregation in PCNs that is computationally feasible, balance secure, optimal, cost efficient, and private
- Future work:
  1. Designing computationally tractable protocol for other topologies
  2. Cross-chain aggregation

Thank you!

myeo@ist.ac.at